

**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects in a computer system from a first storage location to a second storage location, comprising:

[[a))] selecting ~~one or more~~ a data object ~~objects~~ from the first storage location;

[[b))] assigning ~~an identifier (ID)~~ of a first type identifier (ID) to ~~each of the~~ selected data ~~objects~~ object;

[[c))] assigning ~~an ID~~ of a second type ID to ~~each of the~~ selected data ~~objects~~ object;

[[d))] storing the second type ID in a second lock object, indicating that the data object is being accessed;

[[e))] determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object, indicating that the data object is being moved;

[[f))] storing [[a)] the data object, ~~the first ID of which is contained in the first lock object~~, at the second storage location;

[[g))] deleting the data object, ~~the first type ID of which is contained in the first lock object~~, from the first storage location; and

[[h))] deleting the second type ID from the second lock object, indicating that the data object is no longer being accessed, after ~~a particular~~ the first type ID has been stored in the first lock object.

2. (Currently Amended) The method of claim 1, further comprising:

[[i)]] deleting the first type ID from the first lock object, indicating that the data object is no longer being moved, after the respective data object assigned to that at least one ID has been deleted from the first storage location.

3. (Currently Amended) The method of claim 1, wherein the data object comprises one or more fields of one or more tables, and wherein the at least one first type ID and the second type ID comprise ~~comprises~~ one or more key fields of the one or more tables.

4. (Currently Amended) The method of claim 1, wherein the data object is stored in ~~one or more files~~ a file and wherein an assignment of ~~[[a]]~~ the first type ID to the file or to a name of the file, ~~in which the data object assigned to the first type ID is stored,~~ is stored in the first lock object.

5. (Previously Presented) The method of claim 1, wherein the first lock object is stored on a nonvolatile storage means.

6. (Currently Amended) The method of claim 1, wherein the second type ID is stored in the second lock object after ~~the step of assigning an ID of a~~ the second type ID to each of the ~~selected data objects for the respective data object.~~

7. (Currently Amended) The method of claim 1, wherein the second type ~~[[of]] ID of the selected data object~~ is stored in the second lock object before the data object ~~assigned to that ID~~ is stored at the second storage location.

8. (Currently Amended) The method of claim 1, wherein storing the first type ID in ~~[[a]] the~~ first lock object further comprises:

storing the first type IDs of all selected data objects before storing the data object at the second storage location ~~a first storing process according to step f) is started~~.

9. (Currently Amended) The method of claim 1, further comprising:

~~[[j)]]~~ checking whether ~~[[a]] the~~ first type ID ~~for the data object~~ has been stored in ~~[[a]] the~~ first lock object, and if the first type ID ~~for that data object~~ has been stored, skipping at least ~~step f) for that~~ storing the data object at the second storage location.

10. (Currently Amended) The method of claim 1, further comprising:

~~[[k)]]~~ checking whether the data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping at least ~~step f) for that~~ storing the data object at the second storage location.

11. (Currently Amended) The method of claim 10, wherein ~~the~~ checking whether the data object is contained in the second storage location ~~step k)~~ is performed by querying the first lock object.

12. (Currently Amended) The method of claim 1, further comprising:  
~~l) determining whether the data object was stored in the first lock object~~  
~~successfully, and upon a successful storage, checking whether the data object assigned~~  
~~to the respective first ID has been completely stored in the second storage location, and~~  
~~if the respective first ID has not completely been stored, skipping at least steps g) and~~  
~~h) for that data object deleting the data object from the first storage location and deleting~~  
~~the second type ID from the second lock object, and deleting the first type ID from the~~  
first lock object.

13. (Previously Presented) The method of claim 1 for use in an enterprise resource planning software.

14. (Currently Amended) A computer system for processing data, comprising:  
memory means for storing program instructions;  
input means for entering data;  
storage means for storing data;  
a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for moving data

objects in the computer system from a first storage location to a second storage location, the method comprising:

- selecting ~~one or more~~ a data object ~~objects~~ from the first storage location;
- assigning ~~an identifier (ID)~~ of a first type identifier (ID) to ~~each of the selected~~ data ~~objects~~ object;
- assigning ~~an ID~~ of a second type ID to ~~each of the selected~~ data ~~objects~~ object;
- storing the second type ID in a second lock object, indicating that the data object is being accessed;
- determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object, indicating that the data object is being moved;
- storing ~~[[a]]~~ the data object, ~~the first ID of which is contained in the first lock object,~~ at the second storage location;
- deleting the data object, ~~the first type ID of which is contained in the first lock object,~~ from the first storage location; and
- deleting the second type ID from the second lock object, indicating that the data object is no longer being accessed, after a particular the first type ID has been stored in the first lock object.

15. (Canceled).

16. (Currently Amended) A computer readable medium comprising instructions for performing a method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

- selecting ~~one or more~~ a data object objects from the first storage location;
- assigning ~~an identifier (ID)~~ of a first type identifier (ID) to ~~each of the selected~~ data objects object;
- assigning ~~an ID~~ of a second type ID to ~~each of the selected~~ data objects object;
- storing the second type ID in a second lock object, indicating that the data object is being accessed;
- determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object, indicating that the data object is being moved;
- storing ~~[[a]]~~ the data object, ~~the first ID of which is contained in the first lock object,~~ at the second storage location;
- deleting the data object, ~~the first type ID of which is contained in the first lock object,~~ from the first storage location; and
- deleting the second type ID from the second lock object, indicating that the data object is no longer being accessed, after a particular the first type ID has been stored in the first lock object.

17. (Canceled).

18. (Currently Amended) The computer readable medium of claim 16, the method further comprising:

deleting the first type ID from the first lock object, indicating that the data object is no longer being move, after the ~~respective data object assigned to that at least one ID~~ has been deleted from the first storage location.

19. (Currently Amended) The computer readable medium of claim 16, wherein the data object comprises one or more fields of one or more tables, and wherein the ~~at least one~~ first type ID and the second type ID comprise ~~comprises~~ one or more key fields of the one or more tables.

20. (Currently Amended) The computer readable medium of claim 16, wherein the data object is stored in ~~one or more files~~ a file and wherein an assignment of ~~[[a]]~~ the first type ID to the file or to a name of the file, ~~in which the data object assigned to the first type ID is stored,~~ is stored in the first lock object.

21. (Previously Presented) The computer readable medium of claim 16, wherein the first lock object is stored on a nonvolatile storage means.

22. (Currently Amended) The computer readable medium of claim 16, wherein the second type ID is stored in the second lock object after ~~the step of~~ assigning an ID of a the second type ID to each of the ~~selected data objects for the~~ respective data object.

23. (Currently Amended) The computer readable medium of claim 16, wherein the second type ~~[[of]] ID of the selected data object~~ is stored in the second lock object before the data object ~~assigned to that ID~~ is stored at the second storage location.

24. (Currently Amended) The computer readable medium of claim 16, wherein storing the first type ID in ~~[[a]]~~ the first lock object further comprises:  
storing the first type IDs of all selected data objects before storing the data object at the second storage location.

25. (Currently Amended) The computer readable medium of claim 16, the method further comprising:

checking whether ~~[[a]]~~ the first type ID ~~for the data object~~ has been stored in ~~[[a]]~~ the first lock object, and if the first type ID ~~for that data object~~ has been stored, skipping the storing the data object, ~~the first ID of which is contained in the first lock object at the~~ second storage location, ~~for that data object~~.



26. (Currently Amended) The computer readable medium of claim 16, the method further comprising:

checking whether ~~that~~ the data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping the storing the data object, ~~the first ID of which is contained in the first lock object at the second storage location, for that data object.~~

27. (Currently Amended) The computer readable medium of claim ~~[[16]]~~ 26, wherein the checking is performed by querying the first lock object.

28. (Currently Amended) The computer readable medium of claim 16, the method further comprising:

~~determining whether the data object was stored in the first lock object successfully, and upon a successful storage,~~ checking whether the data object assigned to the respective first ID has been completely stored in the second storage location, and if the respective first ID has not completely been stored, skipping the deleting the data object from the first storage location and the deleting the second type ID from the second lock object, ~~after a particular first type ID has been stored in the first lock object for that data object and deleting the first~~ type ID from the first lock object.

29. (Currently Amended) A computerized system for processing data, the computerized system executing a method comprising:

- selecting ~~one or more~~ a data object ~~objects~~ from the a first storage location;
- assigning ~~an identifier (ID)~~ of a first type identifier (ID) to ~~each of the selected~~ data ~~objects~~ object;
- assigning ~~an ID~~ of a second type ID to ~~each of the selected~~ data ~~objects~~ object;
- storing the second type ID in a second lock object, indicating that the data object is being accessed;
- determining whether the second type ID was stored successfully in the second lock object, and upon a successful storage, storing the first type ID in a first lock object, indicating that the data object is being moved;
- storing ~~[[a]]~~ the data object, ~~the first ID of which is contained in the first lock object,~~ at a the second storage location;
- deleting the data object, ~~the first type ID of which is contained in the first lock object,~~ from the first storage location; and
- deleting the second type ID from the second lock object, indicating that the data object is no longer being accessed, after ~~a particular~~ the first type ID has been stored in the first lock object.

30. (Currently Amended) The computerized system of claim 29, the method further comprising:

deleting the first type ID from the first lock object, indicating that the data object is no longer being moved, after the respective data object assigned to that at least one ID has been deleted from the first storage location.

31. (Currently Amended) The computerized system of claim 29, wherein the data object comprises one or more fields of one or more tables, and wherein the ~~at least one~~ first type ID and the second type ID comprise ~~comprises~~ one or more key fields of the one or more tables.

32. (Currently Amended) The computerized system of claim 29, wherein the data object is stored in ~~one or more files~~ a file and wherein an assignment of ~~[[a]]~~ the first type ID to the file or to a name of the file, ~~in which the data object assigned to the first type ID is stored,~~ is stored in the first lock object.

33. (Previously Presented) The computerized system of claim 29, wherein the first lock object is stored on a nonvolatile storage means.

34. (Currently Amended) The computerized system of claim 29, wherein the second type ID is stored in the second lock object after ~~the step of assigning an ID of a~~ the second type ID to each of the ~~selected data objects for the respective data object.~~

35. (Currently Amended) The computerized system of claim 29, wherein the second type ~~[[of]] ID of the selected data object~~ is stored in the second lock object before the data object ~~assigned to that ID~~ is stored at the second storage location.

36. (Currently Amended) The computerized system of claim 29, wherein storing the first type ID in ~~[[a]]~~ the first lock object further comprises:

storing ~~the~~ first type IDs of all selected data objects before storing the data object at the second storage location.

37. (Currently Amended) The computerized system of claim 29, the method further comprising:

checking whether ~~[[a]]~~ the first type ID ~~for the data object~~ has been stored in ~~[[a]]~~ the first lock object, and if the first type ID ~~for that data object~~ has been stored, skipping ~~the~~ storing the data object, ~~the first ID of which is contained in the first lock object~~ at the second storage location, ~~for that data object~~.

38. (Currently Amended) The computerized system of claim 29, the method further comprising:

checking whether the data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping ~~the storing the data object, the first ID of which is contained in the first lock object at the second storage location, for that data object.~~

39. (Currently Amended) The computerized system of claim ~~[[29]]~~ 38, wherein the checking is performed by querying the first lock object.

40. (Currently Amended) The computerized system of claim 29, the method further comprising:

~~determining whether the data object was stored in the first lock object~~ successfully, and upon a successful storage, checking whether the data object assigned to the respective first ID has been completely stored in the second storage location, and if the respective first ID has not completely been stored, skipping ~~the deleting the data object from the first storage location and the deleting the second type ID from the second lock object, after a particular first type ID has been stored in the first lock object for that data object~~ and deleting the first type ID from the first lock object.